KNIFE: A low-cost Reconfigurable system

José Ignacio Hidalgo, Juan Millán-Belsué, Carlos Roa and Juan Lanchares
Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid
28040 Madrid-Spain
hidalgo@dacya.ucm.es

Abstract

This paper presents Knife, a reconfigurable Multi-FPGA based system designed and implemented as a final project to get a degree on electronic engineering. The whole circuit design flow is addressed in order to obtain an automated process. It starts from a hardware description language design specification in VHDL or XNF and targets at a reconfigurable–expandable board, made up of eight Xilinx FPGAs. We have designed and implemented both, the Multi-FPGA board and the software tool. The software tool employs some Xilinx Foundation tools while other steps, such as partitioning and placement of the circuit, uses some ideas proposed in previous papers.

1. Introduction

A reconfigurable or adaptive platform is a system which logic functionality and interconnect can be customized to suit and develop a specific application through post-fabrication, user defined programming [1]. Reconfigurable solutions offer an intermediate solution to ASIC devices, that is, a compromise between the performance and speed of specific hardware devices and the adaptability of software solutions. One of the main reconfigurable devices are those based on Field Programmable Gate Array (FPGA). Multi-FPGA systems (MFSs) are composed of several interconnected FPGA devices and are used as custom computing machines, logic emulators and rapid prototyping vehicles. Although reconfigurable FPGA technologies have been commercially available for over two decade, the number of available tools capable of supporting reconfigurable system design is still insufficient. In addition the implementation (design, compilation and downloading) process for MFS is often complicated because of the large number of tasks, specially for students.

Many such existing tools are based on conventional static FPGA design flows, and demand expert skills and improvisation in order to produce a working reconfigurable system. Relative immaturity of reconfigurable system design tools, together with growing number of reconfigurable technologies, conflicting design requirements, and other factors make the reconfigurable system design a very challenging task. In addition the cost of those systems is sometimes excessive to apply them to small problems or for using MFS during labs of undergraduate courses.

In this paper we present a complete reconfigurable system. We have designed both, the Multi-FPGA board and the software tool. During the implementation process we also use some design steps of the Xilinx Foundation tool and, for the partitioning and placement process, we have applied some techniques presented in previous papers. The objective of this work is to design a low cost reconfigurable system which can be used for medium size applications, while allowing it use for education. Knife, however, has been conceived like an adaptable system to the advance of the devices, that is, that allows in a future, the use of more complex circuits and with greater logical capacity. This is not a problem in our system, since all the methodology is easily expandable to resources with greater benefits.

The rest of the paper is organized as follows. Section 2 explains the design, construction and features of the Multi-FPGA board. Section 3 shows the different parts of the design flow. Section 4 present some experimental results and, finally section 5 remarks some conclusions and future works.

2. Multi-FPGA Board

Our card has been designed considering the results obtained previously with partitioning algorithms presented in a Ph.D Thesis [2]. The card consists of 8 FPGAs of the 4010 family from Xilinx (which can be replaced by other devices of greater capacity and benefits, just adapting some connections, we use those in order to reduce costs and recycle some old components) connected according to a mesh topology. In addition to the FPGAs we have an electrical power supply and some programming lines (DIN, DONE, CCLK, INIT, PROGRAM), which allows the configuration of the board by means of an XChequer cable from Xilinx [3]. The cable transmits the configuration data to all FPGAs within the board, the transmission frequency is 921 kHz. The speed depends on the computer used , in our case with a PC, a Baud Rate of 115200 can be reached. The power supply used is an ATX computer source. This allows us to have the voltages necessary to feed not only the FPGAS, but also the programming cables such as the XChequer. The MFS board also incorporates some jumper pins, for programming and isolation of a group of FPGA within the board. There also six connectors for expansion of the board using other similar card and an Xchequer connection was also implemented. Figure 1 shows the schematic representation of the board.

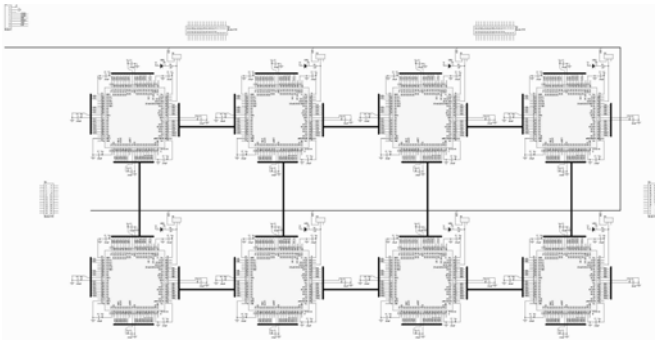

Fig. 1. Schematic of the Multi-FPGA board

3. Design Flow

The whole design flow can be observed on Figure 2. The reader can see how the system presented in this paper, covers all the phases of the design process. Starting off from a specification of the circuit by means of a hardware description language (XNF/VHDL), the final implementation of the circuit over the card previously explained is obtained We will divide the explanation of the design flow in two parts: The formats of the data files and the modules that process them or that modify them.

In order to cover all the design cycle, during the implementation process different formats of the data files are used to describe the circuit. The formats used are XNF, KNF and PNF. XNF is a hardware description language, which was developed by Xilinx. It allows to describe logic circuits in a simple and effective way. In addition, it has the advantage that it is supported by the Xilinx compilation tools. A XNF file is an ASCII text file and it can be also translated from a VHDL description. KNF serves to represent a circuit by means of a text file, using a graph format. This format was created to facilitate the input to the partitioning algorithm of our tool which works on graphs. PNF is an output file which indicates the results of the partitioning and placement processes. For a circuit with N nodes, we obtain a text file of N+1 lines. First, a line with the number of nodes, and then N lines with the information of each node and FPGA assigned.

On the figure six Data Processing Modules of processing can be observed. Read Data, Partitioning and Placement and Rebuild have been developed specifically for our work and the other three, XNF2GND, NGBUILD and MAP are Xilinx design programs. The Read Data program transforms the XNF description of the initial circuit into a graph description. The Partitioning and Placement step uses evolutionary computation techniques to obtain a distribution of the design among the FPGAs, of course we can implement our designs using only part of the 8 devices by means of an manually action. When the partition has been achived we need to re-build the connections between FPGAs parts, this task is the objective of the Rebuild block.

4. Results

Figure 3 show the final implementation of the MFS board. Some designs as an 8-bit adder were successfully implemented and tested on it. During 2005 another student is working for his final-project on the implementation of an asynchronous processor using this board.

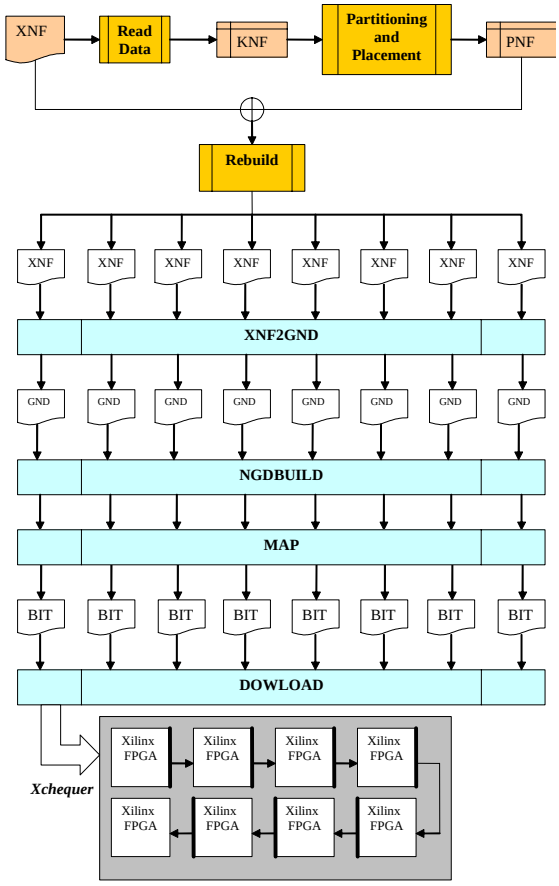

Fig. 2 Design Flow

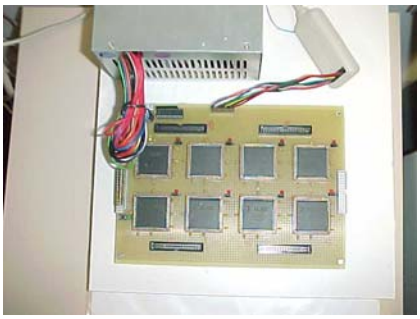

Fig. 3. Final Implementation of the Multi-FPGA board

ACKNOWLEDGMENTS

This work was possible thanks to Spanish Science and Technology Minister project number **TIC 2002/750.**

REFERENCES

[1] R. Tessier and W. Burleson, Reconfigurable Computing and Digital Signal Processing: A Survey, *in Journal of VLSI Signal Processing* , May/June 2001.

[2] http://www.dacya.ucm.es/hidalgo/Publica.htm

[3] *XNF: Xilinx Netlist Format*" http://www.xilinx.com